L Number	Hits	Search Text	DB	Time stamp
2	2	(delet$4 purg$3) with (file$1 document$1 data information) same (log$1 record$3 monitor$3 track$3) same metadata same incremen$5	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:22
3	2	(delet$4 purg$3) with (file$1 document$1 data information) same (log$1 record$3 monitor$3 track$3) same metadata and increment$3 near (backup$1 cop$4 snapshot$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:23
1	103	(delet$4 purg$3) with (file$1 document$1 data information) same (log$1 record$3 monitor$3 track$3) same metadata	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:46
4	176	snapshot with delet$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:47
5	172	restor$5 with snapshot	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:47
7	2339	restor$5 with delet$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 14:32
8	25	(snapshot with delet$3) and (restor$5 with delet$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:48
6	36	(snapshot with delet$3) and (restor$5 with snapshot)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 14:28
9	4	((snapshot with delet$3) and (restor$5 with delet$3)) and @rlad<=20020222	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:49
10	4	((snapshot with delet$3) and (restor$5 with snapshot)) and @rlad<=20020222	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 13:56
11	11	re-delet$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 14:28
12	2	restor$5 with (delet$3 purg$3) near file$1 same incremental near3 backup$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 15:33
13	148	restor$5 with (delet$3 purg$3) near file$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 15:33
14	37	restor$5 with (delet$3 purg$3) near file$1 and @rlad<=20020222	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 15:49

15	1	5974563.pn. and incremental	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2004/05/20 15:49

L Number	Hits	Search Text	DB	Time stamp
1	182	SNAPSHOT WITH DELET$3	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38
2	149	RESTORE$5 WITH SNAPSHOT	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38
3	34	(SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38
4	1935	RESTOR$5 WITH DELET$3	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38
5	19	((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (RESTOR$5 WITH DELET$3)	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38
6	19	((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (RESTOR$5 WITH DELET$3))	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:39
7	10	(((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (RESTOR$5 WITH DELET$3))) AND @AD<20020222	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:40
8	4	(((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (RESTOR$5 WITH DELET$3))) AND @RLAD<20020222	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:41
9	4	(((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (((SNAPSHOT WITH DELET$3) AND (RESTORE$5 WITH SNAPSHOT)) AND (RESTOR$5 WITH DELET$3))) AND @RLAD<=20020222	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:41

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	BRS	L1	182	SNAPSHOT WITH DELET$3	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38	
2	BRS	L2	149	RESTORE$5 WITH SNAPSHOT	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38	
3	BRS	L3	34	1 AND 2	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38	
4	BRS	L4	1935	RESTOR$5 WITH DELET$3	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38	
5	BRS	L5	19	3 AND 4	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:38	
6	BRS	L6	19	3 AND 5	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:39	

	Error Definition	Errors
1		0
2		0
3		0
4		0
5		0
6		0

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
7	BRS	L7	10	6 AND @AD<20020222	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:40	
8	BRS	L8	4	6 AND @RLAD<20020222	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:41	
9	BRS	L9	4	6 AND @RLAD<=20020222	USPAT; US-PGPUB; EPO; IBM_TDB	2004/05/20 13:41	

	Error Definition	Errors
7		0
8		0
9		0

L Number	Hits	Search Text	DB	Time stamp
1	1	restor$3 near3 (file$1 data information document$1) with incremental near5 backup$1 same delet$4 near3 log$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:54
2	1	restor$3 near3 (file$1 data information document$1) same incremental near5 backup$1 same delet$4 near3 log$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:54
3	1	delet$4 near3 (file$1 data information document$1) same incremental near5 backup$1 and (delet$4 purg$3) near3 log$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:56
4	1	(delet$4 purg$3) near3 (file$1 data information document$1) same incremental near5 backup$1 and (delet$4 purg$3) near3 log$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:56
5	1	(delet$4 purg$3) near3 (file$1 data information document$1) same incremental and backup$1 and (delet$4 purg$3) near3 log$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:57
6	1394	restor$3 same (delet$4 purg$3) near5 (file$1 information data document$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:59
7	56	restor$3 same (delet$4 purg$3) near5 (file$1 information data document$1) same log$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 09:59
8	13	restor$3 same (delet$4 purg$3) near5 (file$1 information data document$1) same log$1 and (incremental snapshot$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:44
9	8	restor$3 same (delet$4 purg$3) near5 (file$1 information data document$1) same log$1 and (incremental snapshot$1) and (meta adj data or metadata)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:00
10	5	(restor$3 same (delet$4 purg$3) near5 (file$1 information data document$1) same log$1 and (incremental snapshot$1)) not (restor$3 same (delet$4 purg$3) near5 (file$1 information data document$1) same log$1 and (incremental snapshot$1) and (meta adj data or metadata))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:02
11	2	restor$3 same (file$1 information data document$1) same (incremental snapshot$1) same (delet$4 purg$3) near (log$1 record$3 stack$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:46
12	84	restor$3 same (file$1 information data document$1) same (incremental snapshot$1) same (log$1 record$3 stack$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:47
13	17	restor$3 same (file$1 information data document$1) same (incremental snapshot$1) same (log$1 record$3 stack$3) and (metadata meta adj data)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:56

14	4263	(delet$4 purg$3) with file$1 same (log$1 record$3 monitor$3 track$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:59
15	6	(delet$4 purg$3) with file$1 same (log$1 record$3 monitor$3 track$3) same incremental adj backup$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 10:59
16	25	(delet$4 purg$3 remov$3) with (file$1 document$1 data information) same (metadata) same (log$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/05/20 11:24

TDB-ACC-NO: NNRD430120

DISCLOSURE TITLE: RECONCILING A FILE SYSTEM TO A POINT IN TIME USING INCREMENTAL BACKUP DATA

PUBLICATION-DATA: IBM technical Disclosure Bulletin, February 2000, UK

ISSUE NUMBER: 430

PAGE NUMBER: 337

PUBLICATION-DATE: February 1, 2000 (20000201)

CROSS REFERENCE: 0374-4353-0-430-337

DISCLOSURE TEXT:

An algorithm is described that reconciles a client file system to its state at a specified point in time using incremental backup data stored on a storage-management server. This is accomplished by restoring selected files from the server to the client and by deleting selected files from the client file system. After the algorithm has been executed, the client file system corresponds to its state at the desired point in time. In a typical embodiment, this algorithm is used in a client-server system. A storage-management server stores objects that have been backed up or archived from various client nodes. The server stores client data in a storage hierarchy consisting of various media types (e.g., disk, tape, optical) and uses a database for tracking the attributes and storage location of each client object.

Traditionally, the client and server operate in the following distinct modes. In one mode of operation, the client and server cooperate to perform an incremental backup. In this operation, the client examines a file system and sends to the server a copy of each file that has not previously been backed up to the server or that has been modified since the last backup. For each file received during an incremental backup, the server stores the file in

its storage hierarchy and also stores meta-data about that file in its database. Incremental backup allows each individual file to be cataloged by the server and allows any or all such files to be restored to the server. However, incremental backup and restore can be inefficient because each file must be processed individually by both the client and the server. This inefficiency can be problematic during a restore of an entire file system, since critical time may be lost as each file is restored individually. In an alternate mode of operation, the client and server may perform an image backup. In this operation, the client creates an image of the entire file system and sends this image to the server as a single object. If the entire file system must be restored to the client, the server can send the file system image and the client then processes this image to re-create the file system as it existed at the time the image was created. Image backup and restore operations are efficient because handling of individual files is greatly reduced. However, any changes to the file system after the image was created will not be reflected in the restored file system. In a typical embodiment of this invention as described below, incremental and image backup and restore operations are combined to achieve certain benefits of both operating modes. Specifically, the invention involves restoring a file system image and then reconciling the restored file system so that it matches its state at a specified point in time. This approach affords much of the efficiency inherent in an image restore while still allowing the file system to be restored to the state in which it existed at some point in time subsequent to image creation. The following general steps can be used for restoring a file system to its state at a specified point in time, using both image and incremental backup data residing on a storage management server. A user invokes a client program to request that a file system be restored to its state at some date and time, called toDate.

The client contacts the server and requests a lis t of image backups that have been stored for this client's file system, along with the date on which each image was created. The server obtains this information from its database and sends it to the client. After receiving a list

of images with their creation dates, the client selects the image with the latest creation date prior to toDate. The creation date and time of the selected image is hereafter called fromDate. The client restores the selected image from the server and uses this image to re-create the file system as it existed at fromDate.

The client sends a reconcile query to the server, which is a request for information regarding changes to the client file system (files added, modified, or deleted) after fromDate but before toDate. The server responds to the reconcile query using data obtained during incremental file backups that were performed before and after fromDate. For every file that was added or modified between fromDate and toDate, the server sends a message to the client indicating that a specific file version should be restored; the indicated file version corresponds to the file that existed in the file system at toDate. For every file that was deleted between fromDate and toDate, the server sends a message to the client indicating that the file should be deleted, since the file did not exist in the client file system at toDate.

The client reconciles the file system to its stat e at toDate by performing the indicated restore or delete operation for every restore message or delete message received from the server. The crux of this invention involves step 6 above, entailing the algorithm used by the server in response to a reconcile query to identify file system changes made between fromDate and toDate. Following is a detailed description of how the server does this. The server depends on regular incremental backups to ensure that it has current file backups at any point in time. The reconcile-query algorithm uses several pieces of data that are routinely available for each file version that has been incrementally backed up to the server.

filename - The fully qualified name of the file in t he client's file system. insDate - The date and time on which this version of the file was inserted into the server's database during incremental file backup. deacDate - The date and time at which the version was deactivated on the server. A file version is active when it is first backed up

to the server (deacDate is set to infinity). Subsequently, the file version becomes inactive during an incremental backup if (1) the file has been deleted from the client file system, or (2) a more recent version of the file is backed up because the file has been modified in the client file system since the last backup.

In response to a reconcile query in step 6 above, the server first scans the database entries for inactive backup objects belonging to the specified file system. For each filename, the following steps are performed. Set state variables foundAtFromDate and foundAtToDate to False. If an inactive version is found for which insDate > fromDate (the version is not included in the image) AND insDate < toDate AND deacDate > toDate (the version existed at toDate) then send a restore message to the client for this object. If an inactive version is found for which insDate < fromDate AND deacDate > fromDate then set foundAtFromDate = True.

If an inactive version is found for which insDate < toDate AND deacDate > toDate then set foundAtToDate = True. If the last inactive entry for the filename is found AND the algorithm has not already sent a restore message for this filename AND foundAtFromDate = True AND foundAtToDate = False then check for an active version of this filename. If no active object is found for which insDate < toDate then send a delete message to the client for this filename. The server then scans the database entries for every active backup object belonging to the specified client file system. If an active object is found for which insDate > fromDate AND insDate < toDate then send a restore message to the client for this object.

h c che e e f e